ยืน ภู่วรวรรณ

ZAP ได้ ก็กู้ด้วย FAT ได้

เคยเจอเรื่องแบบนี้ไหม ข้อมูลสูญไปโดยไม่เจตนา คงไม่ต้องพลิกแผ่นดินหา ถ้าลองใช้ FAT กู้แฟ้ม

ภาพประกอบ : สุชาติ สหชาติโกศานันท์

ปัญหาที่เกิดขึ้นกับแฟ้มข้อมูลดีเบสที่สำคัญอีกประการหนึ่งคือการลบข้อมูลในแฟ้มที่เรียกว่า ZAP หรือลบข้อมูลทั้งหมด การลบข้อมูลนี้อาจกระทำโดยไม่ตั้งใจหรือพลั้งเผลอ ซึ่งเป็นปัญหาต่อการใช้งานได้ ดังนั้นจึงมีความจำเป็นที่เราต้องทำความเข้าใจเพื่อแก้ปัญหาที่เกิดขึ้นเมื่อถึงคราวจำเป็น มุมดีเบสในฉบับนี้จึงเป็นวิธีการแก้ปัญหาเกี่ยวกับการลบแฟ้มหรือการลบข้อมูลโดยไม่ตั้งใจ

เมื่อใช้คำสั่ง ZAP

ในที่นี้จะเริ่มทำความเข้าใจกลไกการ ZAP จากจุดเริ่มต้นโดยมีข้อสมมติฐานจากข้อมูลในแฟ้มข้อมูลดังตัวอย่างในรูปที่ 1 แฟ้มที่จะเป็นตัวอย่างเรื่อง ZAP คือแฟ้ม MICRO.DBF ซึ่งมีจำนวนเรคอร์ด 444 เรคอร์ด ขนาดแฟ้ม 32130 กิโลไบต์ เป็นแฟ้มที่มีขนาดใหญ่พอสมควร

เพื่อให้เห็นโครงสร้างการจัดเก็บแฟ้มนี้ในดิสก์ จึงต้องเรียกดูรายละเอียดข้อมูลการจัดเก็บด้วยยูทิลิตี้ จะได้ทราบว่าแฟ้มนี้เก็บไว้ที่คลัสเตอร์อะไรบ้าง โดยใช้ NU เราอาจเรียก NU จากดีเบสก็ได้ (ถ้ามีโปรแกรมอยู่ในดิสก์แล้ว) โดยใช้คำสั่ง RUN NU หาก NU ไม่อยู่ในไดเรคทอรีเดียวกันก็เรียกผ่าน path หรือเส้นทางที่ NU อยู่ก็ได้ ผลบนจอเมื่อเรียก NU แล้ว เราจะเลือก Explore disk ตามด้วยการเลือก Choose item ผลบนจอภาพจะเป็นดังรูปที่ 2 เราเลือกหัวข้อ File จากนั้นเลือก Dir area เมนู

```
Write fault error writing device PRN
Abort, Retry, Ignore, Fail? r

. dir
Database Files    # Records    Last Update     Size
SAMPLE1.DBF               3     09/21/90         391
MICRO.DBF               444     08/26/90       32130
DBASE4.DBF              284     09/08/90       21842
SAMPLE.DBF                3     09/21/90         391

  54754 bytes in     4 files.
  65536 bytes remaining on drive.
```

รูปที่ 1 แฟ้มข้อมูลตัวอย่าง

```
Menu 1.1
                    Choose item

                Change drive

                change Directory

                File

                cLuster

                Sector

                Absolute sector

                Return to Explore disk

              Select a file
Item type | Drive | Directory name                  | File name
Directory |   C:  | \DBASE3                         | Dir area
```

รูปที่ 2 ผลบนจอภาพเมื่อเรียก NU

```
Dir area                                                  Directory format
Cluster 2,824, Sectors 11,403-11,406                      File offset 0, hex 0
                                                                Attributes
Filename Ext    Size     Date      Time     Cluster  Arc R/O Sys Hid Dir Vol
.                        8-17-90   3:02 pm   2824                     Dir
..                       8-17-90   3:02 pm                            Dir
DBASE    SER    18432   7-29-86  12:00 pm   2825   Arc
DBASE    EXE   133632   4-07-87  10:51 am   2834   Arc
DBASE    MSG    12420   4-07-87  10:51 am   2900   Arc
DBASEOLD MSG    12420   4-07-87  10:51 am   2907   Arc
DBASE    OVL   266240   7-24-86  12:00 pm   2914   Arc
DBASEINL OVL    27648   7-29-86  12:00 pm   3044   Arc
SAMPLE1  DBF      391  12-09-90   3:03 pm   1527   Arc
MICRO    DBF    32130   8-26-90  12:28 pm   2344   Arc
DBASE4   DBF    21842   9-08-90  11:12 am   5005   Arc
SAMPLE   DBT            12-09-90   2:36 pm          Arc
SAMPLE   DBF      391   9-21-90   8:23 am   1509   Arc
SAMPLE1  DBT     1024   9-21-90   8:23 am   1533   Arc
SAMPLE   BAK     5363  12-09-90   3:11 pm   1534   Arc
NU       EXE   140736  11-15-88   4:50 pm   1537   Arc
          Filenames beginning with 'σ' indicate erased entries
                        Press Enter to continue
```

รูปที่ 3

จะกลับมาที่เดิม ขั้นตอนต่อไปเราจะเลือก Edit/display item ผลบนจอภาพจะเป็นดังรูปที่ 3 ข้อมูลที่แสดงนี้เป็นข้อมูลไดเรคทอรีของไดเรคทอรีย่อย dbase 3 ซึ่งสิ่งที่เราสนใจขณะนี้คือแฟ้มข้อมูล MICRO.DBF ที่มีขนาด 32130 ไบต์ ปรับปรุงครั้งสุดท้ายเมื่อวันที่ 8-26-90 เวลา 12:28 pm เป็นไฟล์แบบ Arc หรือไฟล์ที่เขียนอ่านได้ ส่วนนี้เรียกว่าไฟล์แอตทริบิวต์

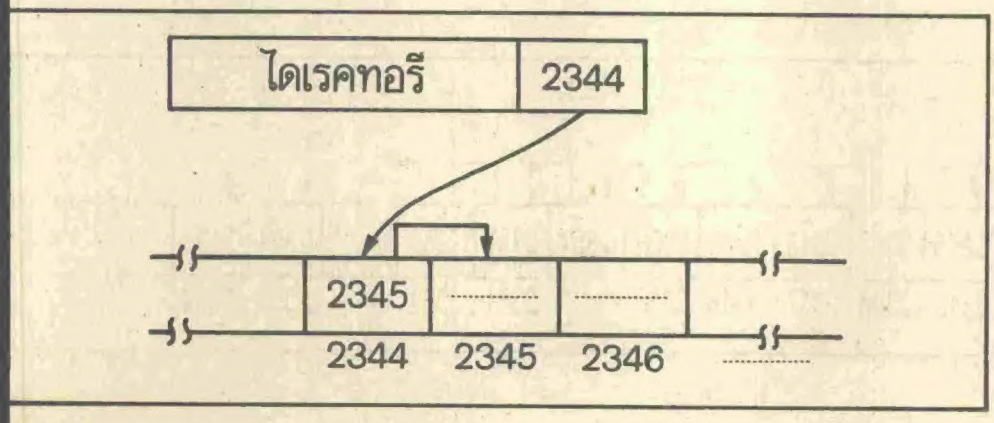
ส่วนสำคัญอีกส่วนหนึ่งคือตัวเลข 2344 เป็นพอยน์เตอร์ที่ชี้ไปยัง FAT หมายเลขคลัสเตอร์ 2344 ซึ่งตีความหมายได้ดังรูปที่ 4

จำนวนคลัสเตอร์กับแฟ้ม

ในการจัดการแฟ้มนั้นผู้เขียนได้กล่าวแล้วว่ามีการจัดข้อมูลเป็นคลัสเตอร์ ถ้าเป็นกรณีของฟลอปปี้ดิสค์คลัสเตอร์หนึ่งจะมีขนาด 2 เซกเตอร์หรือ 1024 ไบต์ แต่ถ้าเป็นฮาร์ดดิสค์ที่มี 4 หัวอ่านจะใช้ 4 เซกเตอร์หรือ 2048 ไบต์เป็นหนึ่งคลัสเตอร์ จากแฟ้มนี้ซึ่งคำนวณจำนวนคลัสเตอร์ได้เป็น

ขนาดแฟ้ม/2048 = จำนวนคลัสเตอร์
(ถ้ามีเศษปัดขึ้นอีกหนึ่ง)

สำหรับแฟ้ม MICRO.DBF เมื่อนำ 32130/2048 จะได้ 16 คลัสเตอร์



รูปที่ 4

```
Menu 1.1.5
                              Select sector

        You may select sectors numbered from 0 through 41,734

                      Starting sector:

                        Ending sector:



                  Outline of Sector Usage on This Disk

              0          Boot area       (used by DOS)
            1 - 82       FAT area        (used by DOS)
           83 - 114      Root Dir. area  (used by DOS)
          115 - 41,734   Data area       (where files are stored)

Item type | Drive | Directory name                          | File name
Directory |   C:  | \DBASE3                                 |  Dir area
```

รูปที่ 5 ผลบนจอภาพหลังจากเลือก sector

หากดูโครงสร้างของคลัสเตอร์ด้วยนอร์ตันยูทิลิตี้สำหรับแฟ้มนี้ โดยเลือกกลับมาที่เมนูของนอร์ตันฯ โดยกด Esc เลือก sector ผลบนจอภาพจะแสดงได้ดังรูปที่ 5

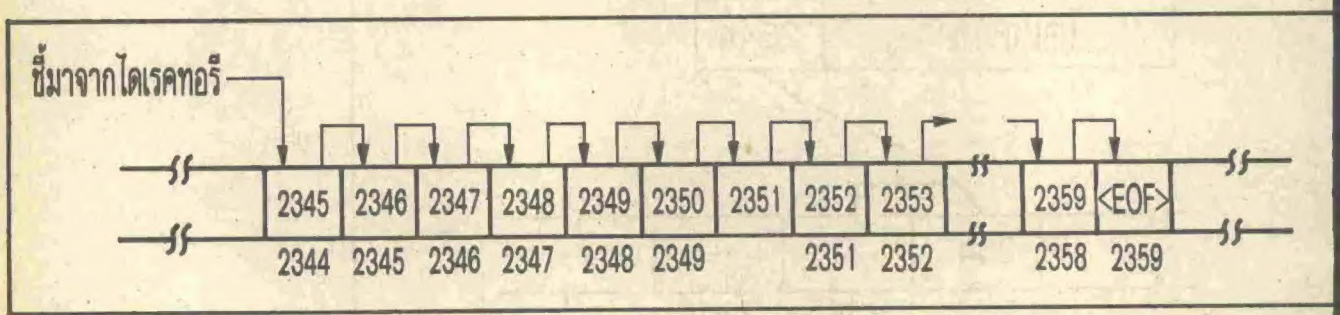
ในกรณีนี้ FAT อยู่ที่เซกเตอร์ 1-82 เราจึงป้อนข้อมูลขอดูเริ่มจากเซกเตอร์ 1 ถึงเซกเตอร์ 82 แล้วเลือก Edit/display data ผลบนจอภาพจะเป็นข้อมูล FAT สิ่งที่น่าสนใจอยู่ที่ FAT ในคลัสเตอร์ที่ 2344 ลองกด PgDn ลงไปเรื่อย ๆ จนมาถึงคลัสเตอร์ 2344 จะได้รายละเอียดดังรูปที่ 6 ที่คลัสเตอร์ 2344 มีตัวเลขชี้ไป 2345 ตัวเลขทั้งหมดจะเป็นดังรูปที่ 7 สังเกตได้ว่าการเชื่อมโยงคลัสเตอร์

```
Sector 1-82                                                    FAT format
Sector 10 in 1st copy of FAT                          Cluster 2,304, hex 900

 2305  2306  2307  2309  3076  2310  2311  2312  2313  2314  2315  2316
  2317  2318  2319  2321 <EOF>  2322  2323  2324  2325  2326  2327  2328
  2329  2330  2331  2332  2333  2334  2335  2336  2337 <EOF>  2339  2340
  2341  2342 <EOF>  2360  2345  2346  2347  2348  2349  2350  2351  2352
  2353  2354  2355  2356  2357  2358  2359 <EOF>  2361 <EOF>  2363  2365
 <EOF>  2369  2367 <EOF> <EOF>  2377  2371  2372  2373  2374  2375  2376
  2394  2378  2379  2380  2381  2382  2383  2384  2385  2386  2387  2388
  2389  2390  2391  2392  2393  2409  2395  2396  2397  2398  2399  2400
  2401  2402  2403  2404  2405  2406  2407  2408  3075  2410  2411  2412
  2413  2414  2415  2416  2417  2418  2419  2420  2421  2422  2423  2424
  2425  2426  2427  2428  2429  2430  2431  2432 <EOF>  2434  2435  2436
  2437  2438  2439  2440  2441  2442  2443  2444  2445  2446  2447  2448
  2449  2450  2451  2452  2453  2454  2455  2456  2457  2458  2459  2460
  2461  2462  2463 <EOF> <EOF>  2466  2467 <EOF>  2469  2470  2471  2472
  2473 <EOF> <EOF>  2476  2477 <EOF>  2479  2480 <EOF>  2482  2483 <EOF>
  2485  2486 <EOF>  2488  2489 <EOF>  2491  2492  2493  2494  2495  2496
  2497  2498  2499  2500  2501  2502  2503  2504  2505  2506  2507  2508
  2509  2510  2511  2512  2513  2514  2515  2516  2517  2518  2519  2520
  2521  2522  2523  2524  2525  2526  2527  2528  2529  2530  2531  2532

                        Press Enter to continue
```

คลัสเตอร์ 2344

จบแฟ้ม

รูปที่ 6 รายละเอียดข้อมูล FAT



รูปที่ 7

ทั้ง 16 คลัสเตอร์นี้จะเชื่อมต่อกัน ไม่กระโดดไปมา ดังนั้นข้อมูลที่เก็บในแฟ้ม MICRO.DBF จึงเรียงกันจากคลัสเตอร์ 2344 ถึงคลัสเตอร์ 2359

เมื่อเจอ ZAP อะไรจะเกิดขึ้น

ย้อนกลับมาที่ดีเบสโดยเรียกแฟ้ม MICRO.DBF มาใช้โดยพิมพ์คำสั่ง use micro ดังรูปที่ 8 ครั้นเมื่อถูก ZAP ผลคือดีเบสจะเปลี่ยนข้อมูลส่วนเฮดเดอร์ในโครงสร้างโดยลดจำนวนเรคอร์ดลงเหลือ 0 เรคอร์ด และวันที่ในการปรับปรุงเป็น 12/23/90 ขนาดความยาวของเรคอร์ดเป็น 72 ไบต์

เมื่อใช้คำสั่ง ZAP ดีเบสจะกระทำดังนี้

```
. use micro
. list stru
Structure for database: C:micro.dbf
Number of data records:     444
Date of last update    : 08/26/90
Field  Field Name  Type       Width    Dec
    1  WORD        Character     33
    2  P1          Character      4
    3  WORD2       Character     30
    4  P2          Character      4
** Total **                     72

  display
Record#  WORD                          P1   WORD2
2
      1  CPU                           9    READY
0
. zap
Zap C:micro.dbf? (Y/N) Yes
. list stru
Structure for database: C:micro.dbf
Number of data records:      0
Date of last update    : 12/23/90
Field  Field Name  Type       Width    Dec
    1  WORD        Character     33
    2  P1          Character      4
    3  WORD2       Character     30
    4  P2          Character      4
** Total **                     72
```

รูปที่ 8 แฟ้ม MICRO.DBF ที่เรียกออกมาใช้งาน

1. เปลี่ยนตัวเลขจำนวนเรคอร์ดที่เฮดเดอร์ของแฟ้มให้เหลือ 0

2. เขียนรหัส 1A ซึ่งเป็นรหัส EOF ลงในไบต์แรกหลังจากจบส่วนเฮดเดอร์

3. ทำการเคลียร์ส่วนของ FAT ให้เป็น 0 เพื่อทำเป็นคลัสเตอร์ว่าง

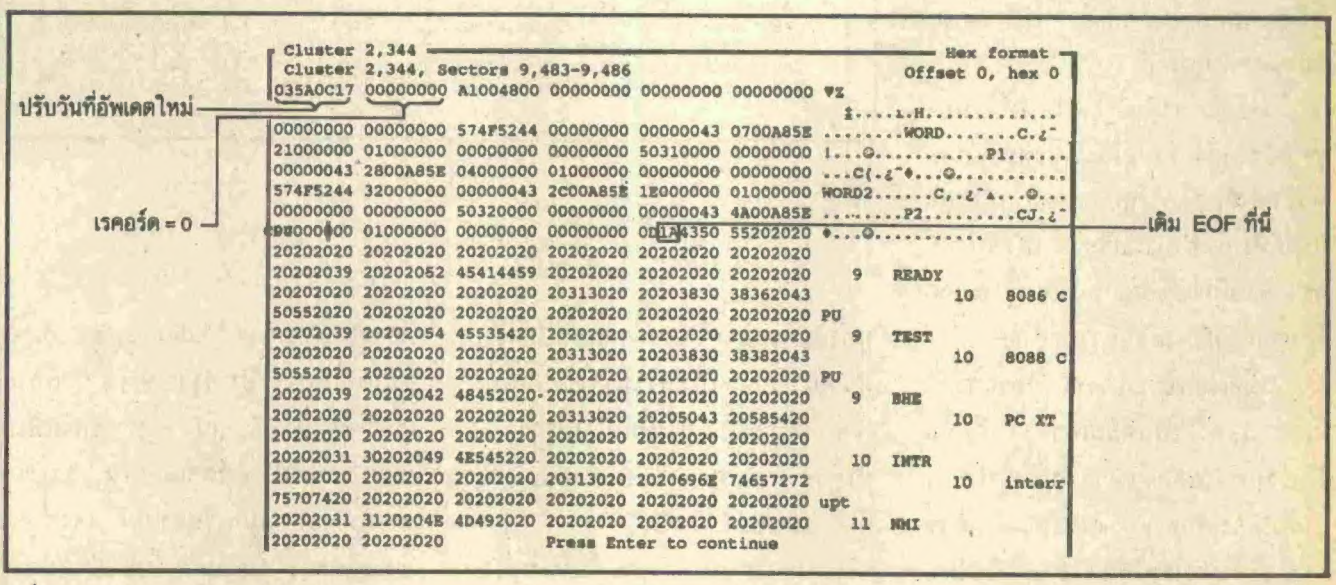
4. เปลี่ยนขนาดของแฟ้มในไดเรคทอรีให้เหลือเพียงแค่เฮดเดอร์เท่านั้น

ดังนั้นการเรียกข้อมูลคืนจะต้องกระทำทั้ง 4 ส่วนนี้

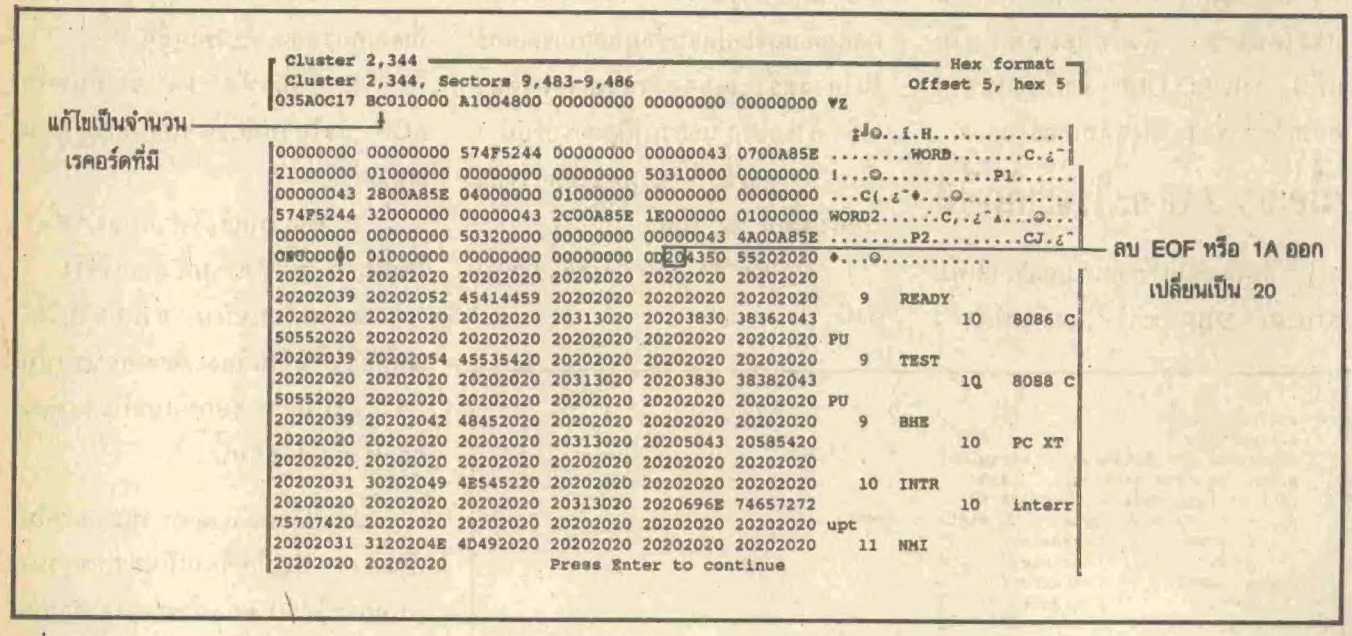
1. เปลี่ยนตัวเลขจำนวนเรคอร์ดให้เป็น 444 ซึ่งเมื่อเขียนเป็นตัวเลขฐานสิบหกจะได้ 01 B6 เท่ากับ 444 ตัวเลขฐานสิบ เราแก้ไขได้โดยใช้ NU ดังนี้

– เรียก NU เลือก Explore disk เลือก Choose item เลือก cluster จากนั้นใส่คลัสเตอร์ที่เรารู้แล้วคือ 2344 หากเรายังไม่รู้จะต้องเรียกดูจากไดเรคทอรีของแฟ้มนี้ว่าชี้มาที่คลัสเตอร์ใด หลังจากนั้นเลือก Edit/display data ผลบนจอจะเป็นดังรูปที่ 9

– สังเกตว่าข้อมูล 03 คือดีเบสทรีเป็นลายเซ็นของแฟ้ม SA คือ ปี ค.ศ.



รูปที่ 9



รูปที่ 10

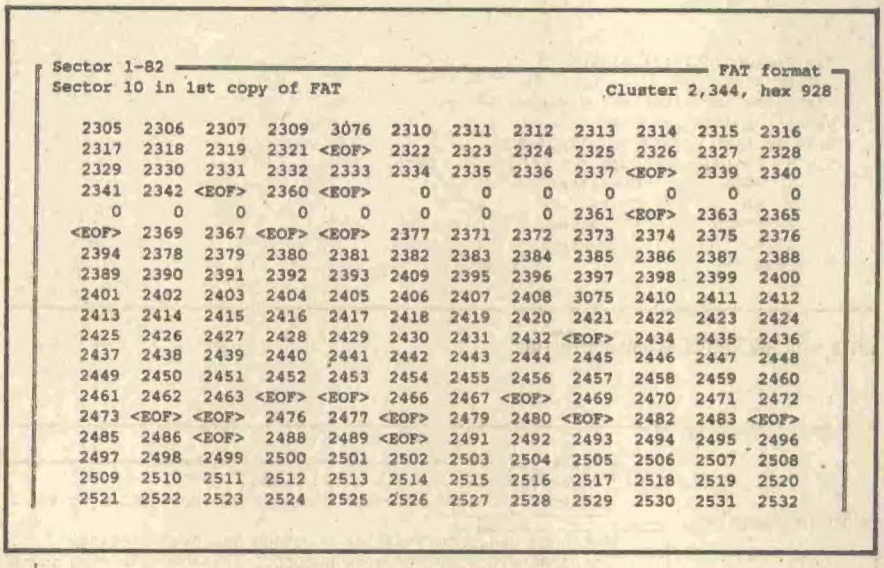
1990 คือตัว 90 ในเลขฐานสิบหก 0C คือเดือน 12 และ 17 ตัวเลขฐานสิบหกคือ 23 หรือวันที่ 23 สำหรับไบต์ถัดมาคือ 00 00 00 ส่วนนี้คือจำนวนเรคอร์ดให้ป้อน BC 01 00

2. ในไบต์ 161 จะมีรหัส 1A เป็นรหัส EOF ให้แก้ไขเป็น 20 หมายถึงรหัสแบลงก์ ผลจะเป็นดังรูปที่ 10

เมื่อแก้ไขสองส่วนนี้เสร็จแล้ว ก็เขียนกลับลงในดิสก์ ด้วยคำสั่ง Write ของนอร์ตันยูทิลิตี้

3. ทำการต่อ FAT ให้ครบทั้ง 16 คลัสเตอร์ ซึ่งปกติควรจะตรวจสอบทีละคลัสเตอร์ โดยขอดูว่าเป็นข้อมูลของแฟ้มที่ต้องการหรือไม่ เพราะบางครั้งหากมีการชี้คลัสเตอร์แบบกระโดดไปมาจะได้เรียงต่ออย่างถูกต้อง

เมื่อเรียกข้อมูล FAT มาดูจะพบว่ามีเลข 0 หรือเป็นคลัสเตอร์ว่างเกิดขึ้นในส่วนคลัสเตอร์ของแฟ้มดังรูปที่ 11 เมื่อเปลี่ยนตัวเลข <EOF> เพื่อชี้ต่อเนื่องกันไปโดยใส่ 2345 แล้วเรียงกันไปจนถึง <EOF> ส่วน EOF นี้



รูปที่ 11

4. เรียงขนาดของแฟ้มในไดเรคทอรี โดยเลือก File แล้วเลือก Dir area นอร์ตันฯจะบอกให้กดตัวอักษร E ลงไปตัวเดียว ผลจะเป็นดังรูปที่ 12 จากนั้นเขียนกลับลงในดิสก์เพื่อปรับปรุง FAT ที่เชื่อมต่อข้อมูลที่เขียนไว้แล้วเพื่อทำการโยงแฟ้มเข้าหากัน

ต่อจากนั้นทำการ Edit/display data ผลบนจอภาพเป็นดังรูปที่ 13 เลื่อนเคอร์เซอร์มาที่ตัวเลข 512 ของแฟ้ม MICRO.DBF แก้ตัวเลขเป็น 32130 จากนั้นเขียนกลับลงไปยังดิสก์ ลองเรียกดีเบสอีกครั้ง เมื่อเรียกดูด้วยไดเรคทอรี ผลที่ได้จะเป็นดังรูปที่ 14

```
Sector 1-82                                                    FAT format
Sector 10 in 1st copy of FAT                          Cluster 2,359, hex 937

  2305  2306  2307  2309  3076  2310  2311  2312  2313  2314  2315  2316
  2317  2318  2319  2321 <EOF>  2322  2323  2324  2325  2326  2327  2328
  2329  2330  2331  2332  2333  2334  2335  2336  2337 <EOF>  2339  2340
  2341  2342 <EOF>  2360  2345  2346  2347  2348  2349  2350  2351  2352
  2353  2354  2355  2356  2357  2358  2359 <EOF>  2361 <EOF>  2363  2365
 <EOF>  2369  2367 <EOF> <EOF>  2377  2371  2372  2373  2374  2375  2376
  2394  2378  2379  2380  2381  2382  2383  2384  2385  2386  2387  2388
  2389  2390  2391  2392  2393  2409  2395  2396  2397  2398  2399  2400
  2401  2402  2403  2404  2405  2406  2407  2408  3075  2410  2411  2412
  2413  2414  2415  2416  2417  2418  2419  2420  2421  2422  2423  2424
  2425  2426  2427  2428  2429  2430  2431  2432 <EOF>  2434  2435  2436
  2437  2438  2439  2440  2441  2442  2443  2444  2445  2446  2447  2448
  2449  2450  2451  2452  2453  2454  2455  2456  2457  2458  2459  2460
  2461  2462  2463 <EOF> <EOF>  2466  2467 <EOF>  2469  2470  2471  2472
  2473 <EOF> <EOF>  2476  2477 <EOF>  2479  2480 <EOF>  2482  2483 <EOF>
  2485  2486 <EOF>  2488  2489 <EOF>  2491  2492  2493  2494  2495  2496
  2497  2498  2499  2500  2501  2502  2503  2504  2505  2506  2507  2508
  2509  2510  2511  2512  2513  2514  2515  2516  2517  2518  2519  2520
  2521  2522  2523  2524  2525  2526  2527  2528  2529  2530  2531  2532
```

รูปที่ 12

```
Dir area                                                  Directory format
Cluster 2,824, Sectors 11,403-11,406                     File offset 0, hex 0
                                                              Attributes
Filename Ext    Size     Date      Time     Cluster  Arc R/O Sys Hid Dir Vol
--------.---  --------  --------  --------  --------  ---------------------
.                        8-17-90   3:02 pm    2824                    Dir
..                       8-17-90   3:02 pm                            Dir
DBASE    SER     18432   7-29-86  12:00 pm    2825   Arc
DBASE    EXE    133632   4-07-87  10:51 am    2834   Arc
DBASE    MSG     12420   4-07-87  10:51 am    2900   Arc
DBASEOLD MSG     12420   4-07-87  10:51 am    2907   Arc
DBASE    OVL    266240   7-24-86  12:00 pm    2914   Arc
DBASEINL OVL     27648   7-29-86  12:00 pm    3044   Arc
SAMPLE1  DBF       391  12-09-90   3:03 pm    1527   Arc
MICRO    DBF       512  12-23-90  12:31 pm    2344   Arc
DBASE4   DBF     21842   9-08-90  11:12 am    5005   Arc
SAMPLE   DBT            12-09-90   2:36 pm           Arc
SAMPLE   DBF       391   9-21-90   8:23 am    1509   Arc
SAMPLE1  DBT      1024   9-21-90   8:23 am    1533   Arc
SAMPLE   BAK      5363  12-09-90   3:11 pm    1534   Arc
NU       EXE    140736  11-15-88   4:50 pm    1537   Arc
--------.---  --------  --------  --------  --------  ---------------------
          Filenames beginning with 'σ' indicate erased entries

                    Press Enter to continue
```

รูปที่ 13

```
. dir
Database Files    # Records    Last Update     Size
SAMPLE1.DBF             3     09/21/90         391
MICRO.DBF             444     12/23/90       32130
DBASE4.DBF            284     09/08/90       21842
SAMPLE.DBF              3     09/21/90         391

  54754 bytes in     4 files.
  65536 bytes remaining on drive.
```

รูปที่ 14 แฟ้ม MICRO.DBF ที่เรียกดูด้วยไดเรคทอรีหลังจากแก้ไขแล้ว

วิธีแก้ปัญหา เมื่อลบแฟ้มข้อมูล

ปัญหาที่เกิดขึ้นกับแฟ้มข้อมูลหรือแฟ้มใด ๆ อีกประการหนึ่งคือการ del หรือ erase ซึ่งจะทำให้แฟ้มหายไปจากไดเรคทอรี การลบแฟ้มที่กระทำในดอสหรือในโปรแกรมใดก็ตามมีกลไกการทำงานดังนี้

1. ดอสจะเปลี่ยนชื่อแฟ้มโดยให้ไบต์แรกเป็น E5 ดังนั้นเมื่อเวลาขอดูด้วย dir ก็จะเห็นเป็นแฟ้มที่ลบแล้ว

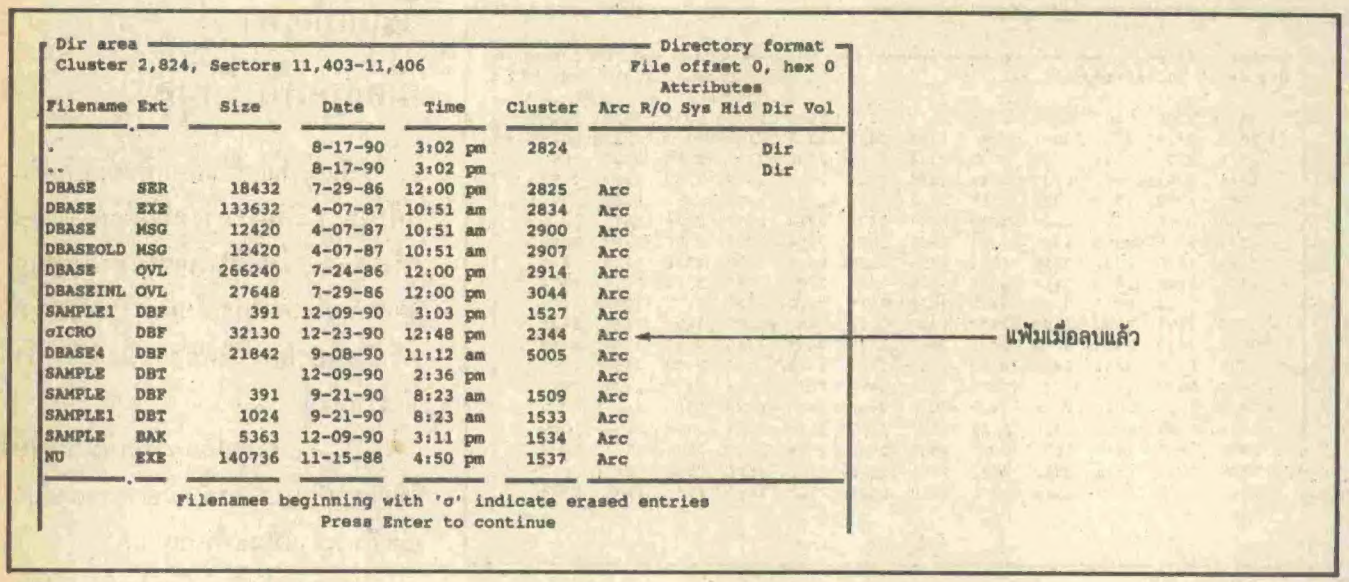
2. ดอสจะไปเคลียร์ในส่วน FAT ให้มีค่าเป็น 0 หมด เพื่อให้เห็นเป็นคลัสเตอร์ว่าง

วิธีการแก้ไขจึงทำได้ง่ายกว่าการ ZAP ลองดูวิธีการโดยเริ่มจากการลบแฟ้ม MICRO.DBF ออกจากดิสค์ดังรูปที่ 15 เมื่อรอดูไดเรคทอรีจะพบรายละเอียดดังรูปที่ 16 สังเกตว่าส่วนของไดเรคทอรี ณ ตำแหน่งชื่อแฟ้มในไบต์แรกจะมีการเปลี่ยนข้อมูล โดยมีรหัส E5 เข้ามาแทน ส่วนอื่นจะยังคงเหมือนเดิมหมด และเมื่อขอดู FAT จะปรากฏเป็นคลัสเตอร์ว่างเริ่มจาก 2344 เป็นต้น

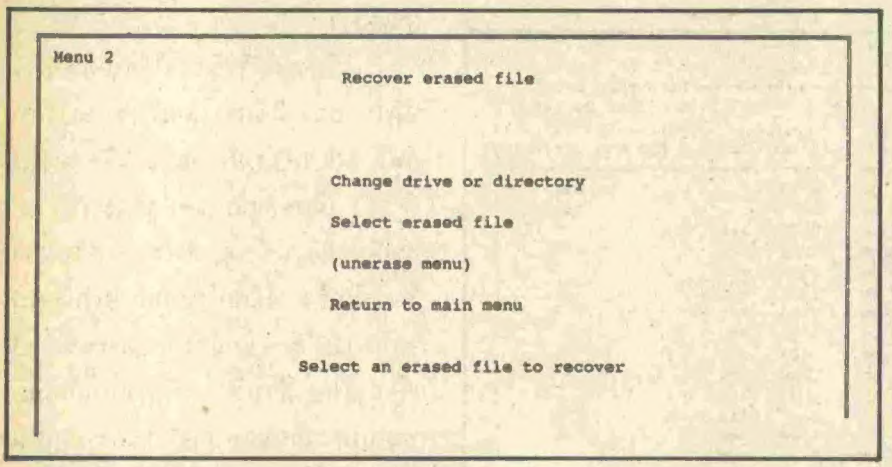
FAT ตรงส่วน 2344 เป็นต้นไปจะปรากฏตัวเลข 0 ซึ่งหมายถึงเป็นคลัสเตอร์ว่าง การแก้ไขจำต้องหาทางเชื่อมต่อ FAT เหล่านี้เข้าด้วยกัน ในนอร์ตันยูทิลิตี้มีโปรแกรมช่วยแก้ปัญหาเรื่องการกู้แฟ้มได้โดยตรง ซึ่งจะใช้เมนูแรกที่มีคำว่า unerase เมื่อเลือก

```
. use
. dele file micro.dbf
File has been deleted.
```

รูปที่ 15



รูปที่ 16

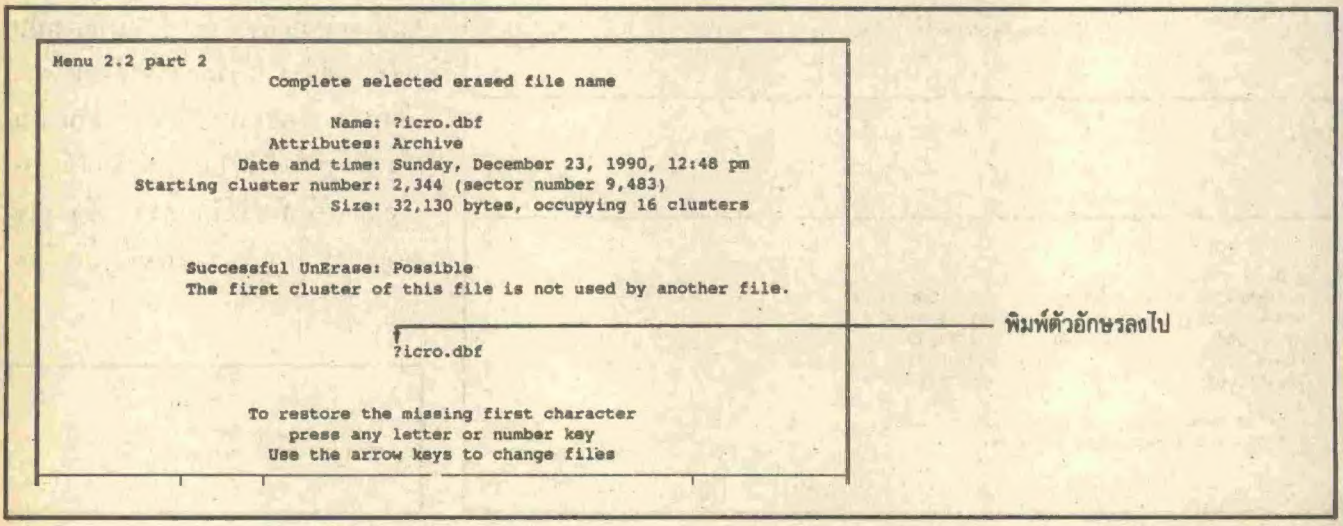


รูปที่ 17 รายละเอียดเมื่อเราเลือก Unerase

Unerase จะปรากฏรายละเอียดดังรูปที่ 17

– เลือก Select erased file นอร์ตันฯจะเลือกแฟ้มที่ลบไปแล้วขึ้นมาให้ โดยมีเครื่องหมาย? อยู่หน้า เราเลือกแฟ้ม ?icro.dbf โดยเติมตัวอักษร m ลงไปในช่อง? ดูรูปที่ 18 ประกอบ หลังจากนั้นกลับเข้าไปยังเมนู unerase ซึ่งผลบนจอภาพจะเป็นดังรูปที่ 19

– เลือกการ Add new cluster ซึ่งนอร์ตันฯจะหาคลัสเตอร์มาใส่ให้ โดยดูการเรียงกันจากตัวชี้ในไดเรคทอรี-



รูปที่ 18

พอยน์เตอร์ แล้วกดรีเทอร์นจะไปยัง All cluster automatically นอร์ดันฯ จะหาข้อมูลคลัสเตอร์มาเรียงต่อ ผลบนจอภาพจะเป็นดังรูปที่ 20

เมื่อได้จำนวนคลัสเตอร์ทั้งหมดแล้วก็จัดการเก็บลงแฟ้มโดยใช้เมนูคำสั่ง save ผลก็คือจะได้ข้อมูลกลับมาครบทุกประการ

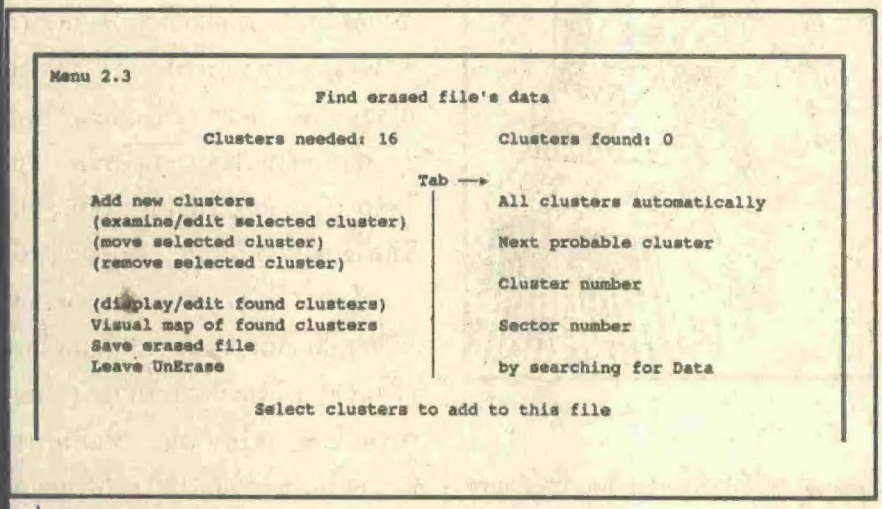
การกู้แฟ้มที่ลบแล้วมีขั้นตอนง่ายกว่า ZAP คือ

1. ให้เปลี่ยนรหัส E5 ในไดเรคทอรีจากส่วนชื่อแฟ้มเป็นตัวรหัสแอสกีตามชื่อแฟ้มที่ต้องการ

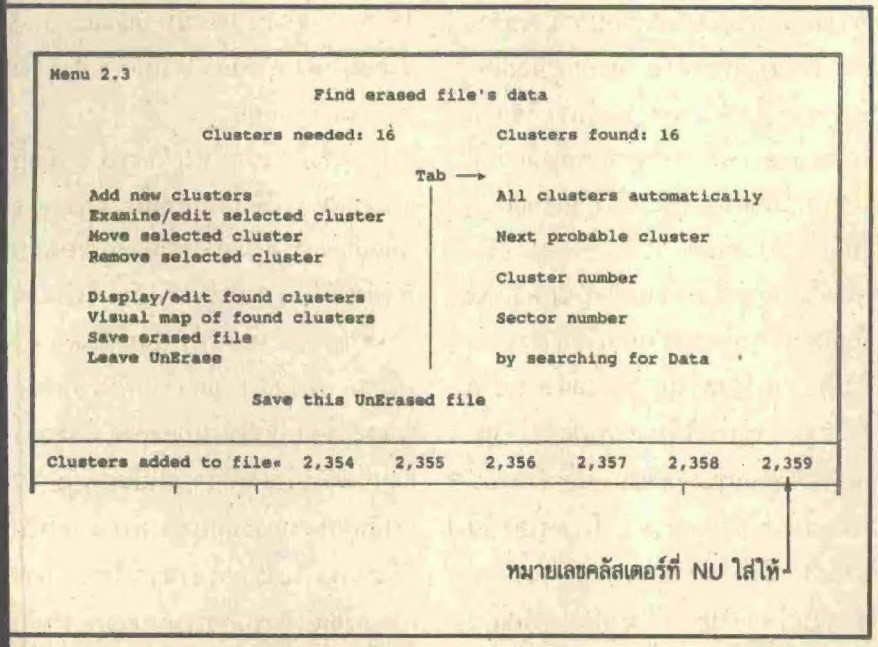
2. จัดการต่อคลัสเตอร์ใน FAT ซึ่งอาจสั่งนอร์ดันฯทำให้อย่างอัตโนมัติหรือต่อเองทีละคลัสเตอร์ก็ได้ จนกว่าจะได้คลัสเตอร์ครบทุกคลัสเตอร์

ส่งท้าย

การจัดการกู้แฟ้มในกรณีนี้ใช้ได้กับแฟ้มทุกประเภท ดังนั้นเราต้องเข้าใจการเชื่อมโยงของ FAT ให้ดี เพราะ FAT สามารถประยุกต์ในเรื่องการแก้ปัญหาต่าง ๆ ได้



รูปที่ 19



รูปที่ 20 นอร์ดันฯจะหาข้อมูลคลัสเตอร์มาเรียงต่อ